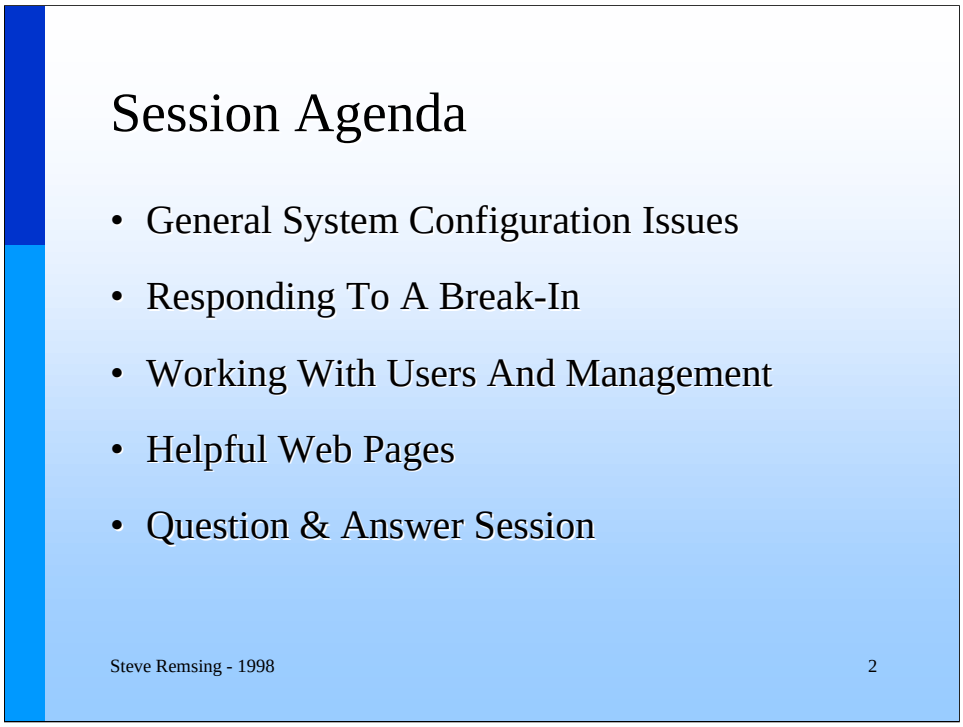
Dealing With Your First Break-in:

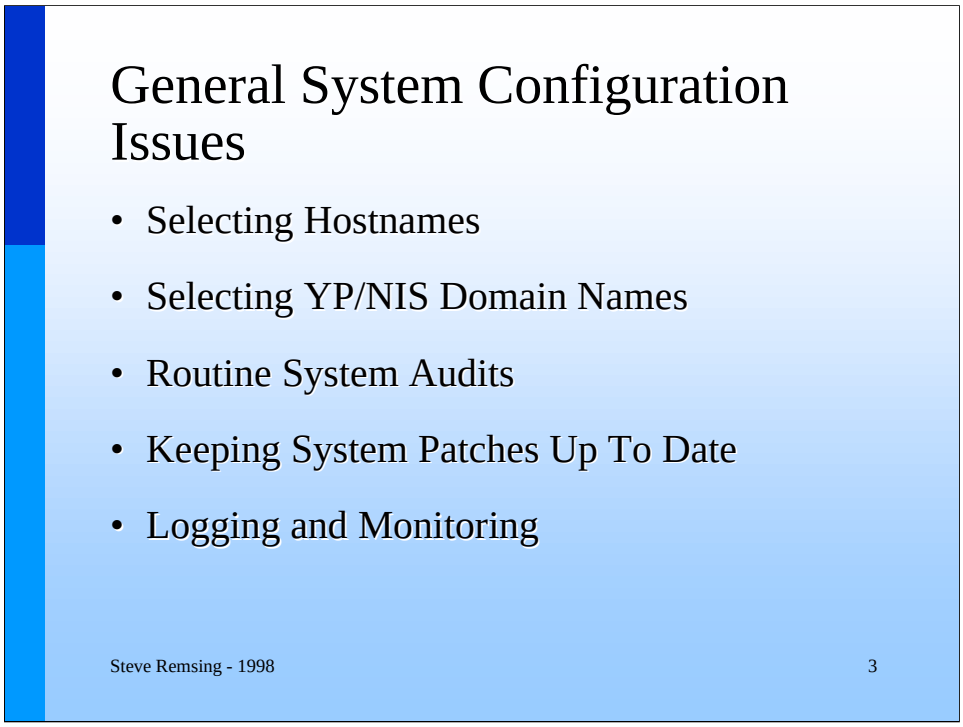
Mistakes Made, Lessons Learned

Steve Remsing
Raytheon STX
Laboratory For High Energy Astrophysics
NASA/Goddard Space Flight Center

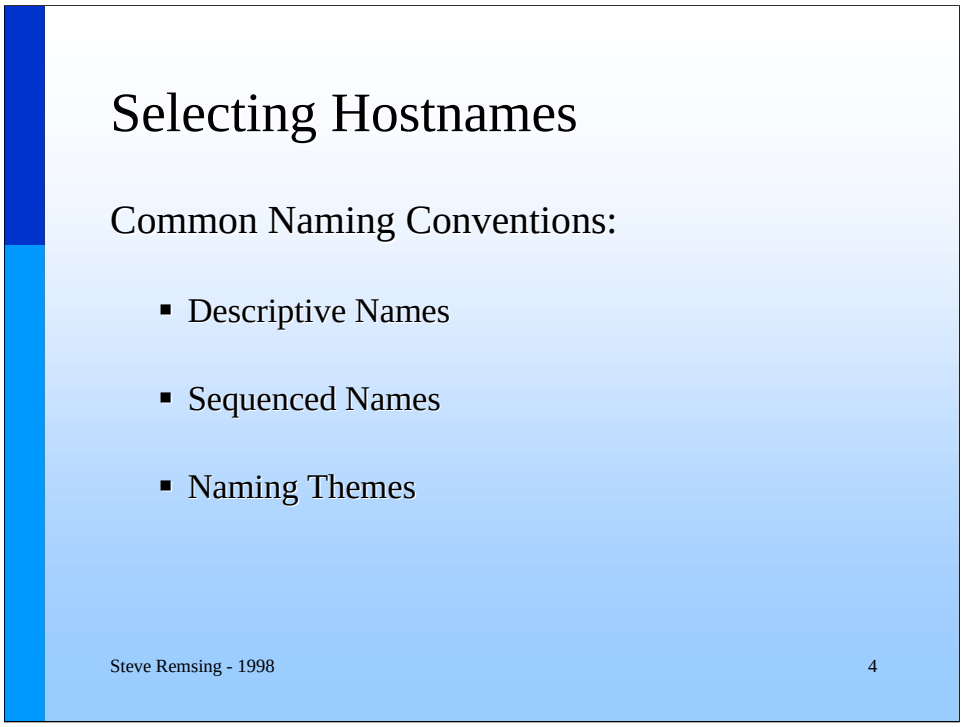


Notes:

There are a number of things you can do now, before a break-in occurs, to prevent a break-in, or at least make it easier to detect and investigate. Even with the most careful configuration, you should always be prepared to deal with a break-in at anytime.

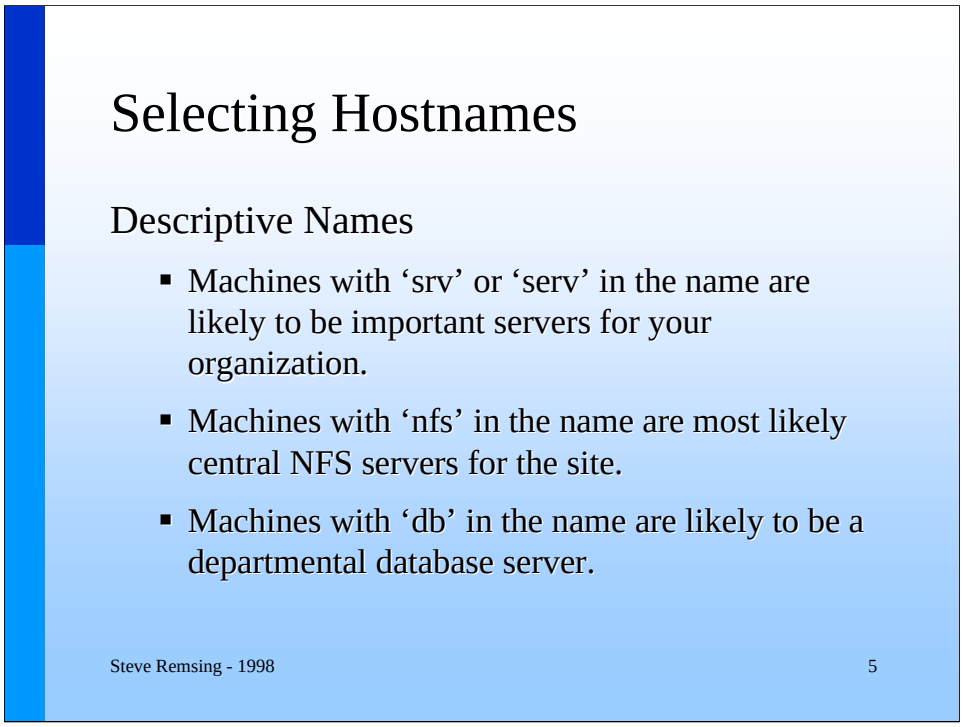

General System Configuration Issues
• Selecting Hostnames
• Selecting YP/NIS Domain Names
• Routine System Audits
• Keeping System Patches Up To Date
• Logging and Monitoring
Steve Remsing - 1998
3

Notes:



Notes:

What's in a name? System names often convey the function or importance of the machine. This is information you don't want a hacker to have as they will use it against you.

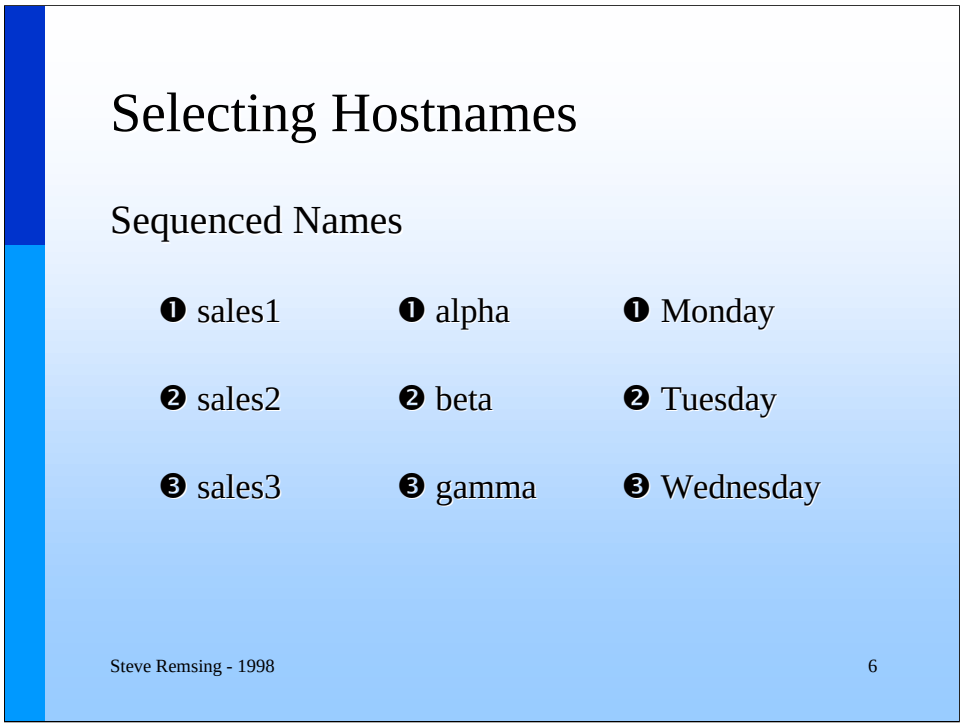


Notes:

While descriptive names are useful, they reveal a good deal of information to the outside world.

All the machines in the above example would be a great target for a hacker for either a break-in or denial of service (DoS) attack.

Consider using generic names for the actual hostname and using aliases only visible internally for all descriptive names.

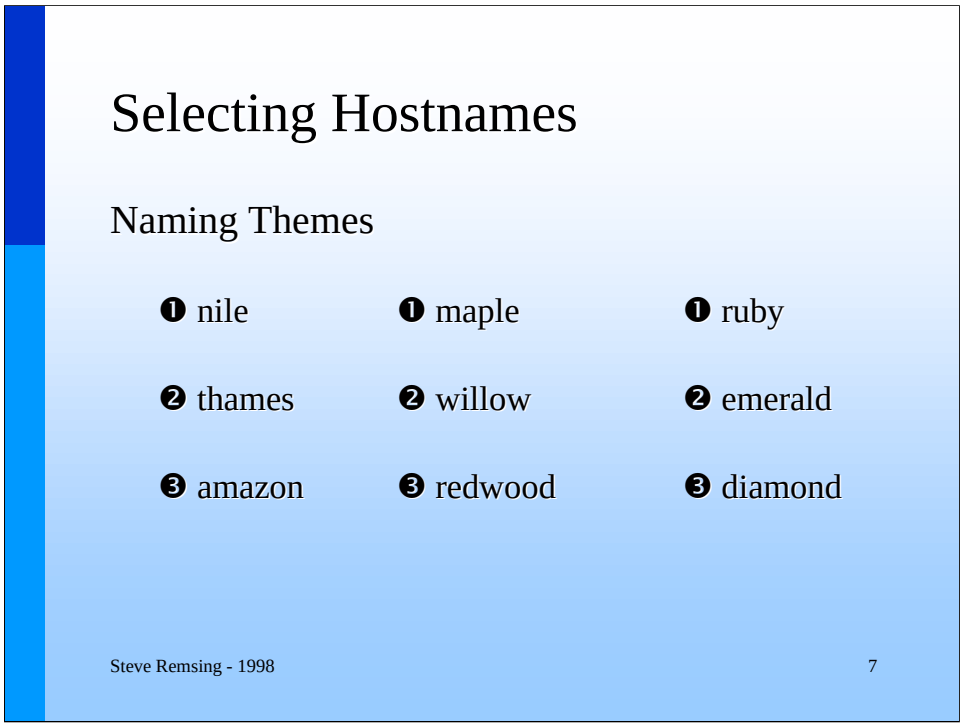


Notes:

Names of this nature allow an intruder to infer the existence of additional hosts, hosts that are likely to be related and therefore possibly trusted by each other.

The “sales” sequence above is also descriptive of the function of these machines, or at least the department they are located in.

Hackers can use this kind of information directly to attack additional machines or indirectly in a social engineering attack to gain more information.



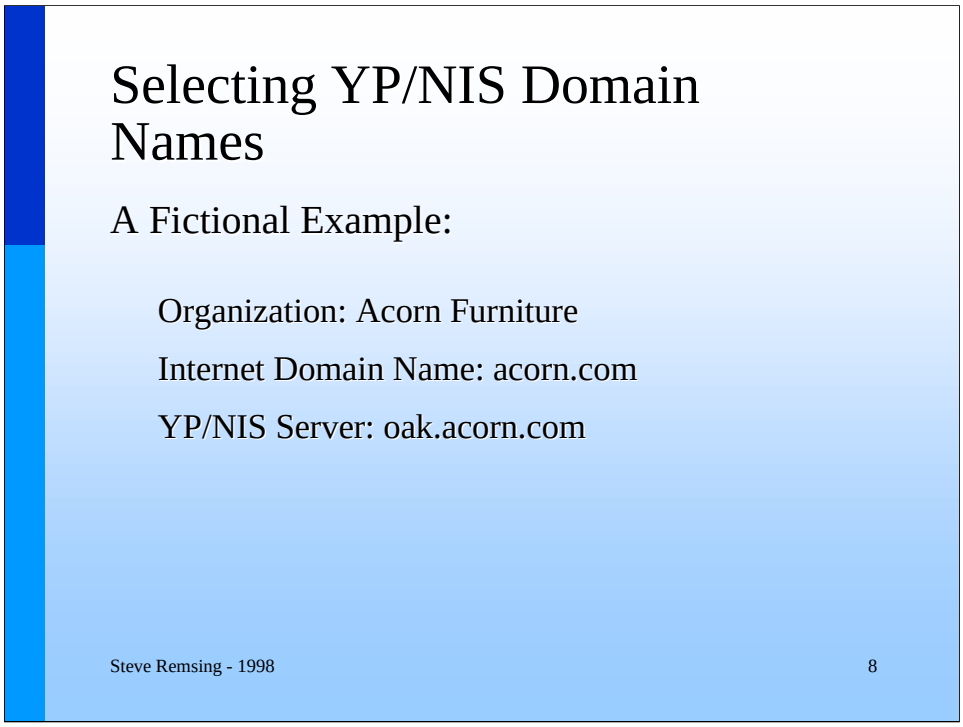
Notes:

There is an important difference between naming themes and sequenced names. The ease of predicting additional names in a theme is more difficult than a sequence.

In the sequence naming example, the existence of “sales1” suggests that there is at least a “sales2” machine and possibly more.

In the naming theme example, the existence of “nile” does not imply the existence of “thames”. A hacker might guess that there are more machines named after rivers but they will have to guess at the names.

Naming themes are not perfect. With some thought it is still possible to guess some additional names in the theme. Machines named with a theme are still likely to be related and therefore may have trust relationships that can be exploited as with sequenced names.

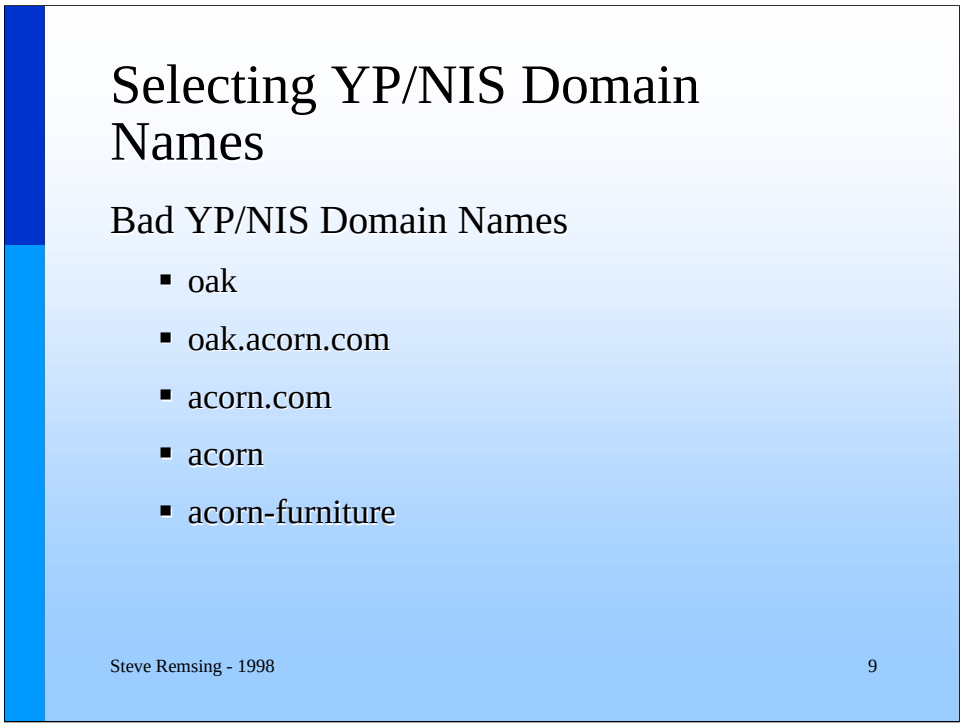


Notes:

The convenience of using the YP/NIS service to provide a centralized password database comes at a price: weak security. The security of YP/NIS is based on knowledge of the domain name. If you know the domain name of a server you can remotely display the contents of any YP/NIS file, including the password file.

A publicly available package called YPX will attempt to guess the domain name of a server and display the password file. This program can be used to display any file, it just defaults to the password file. You can find YPX at:

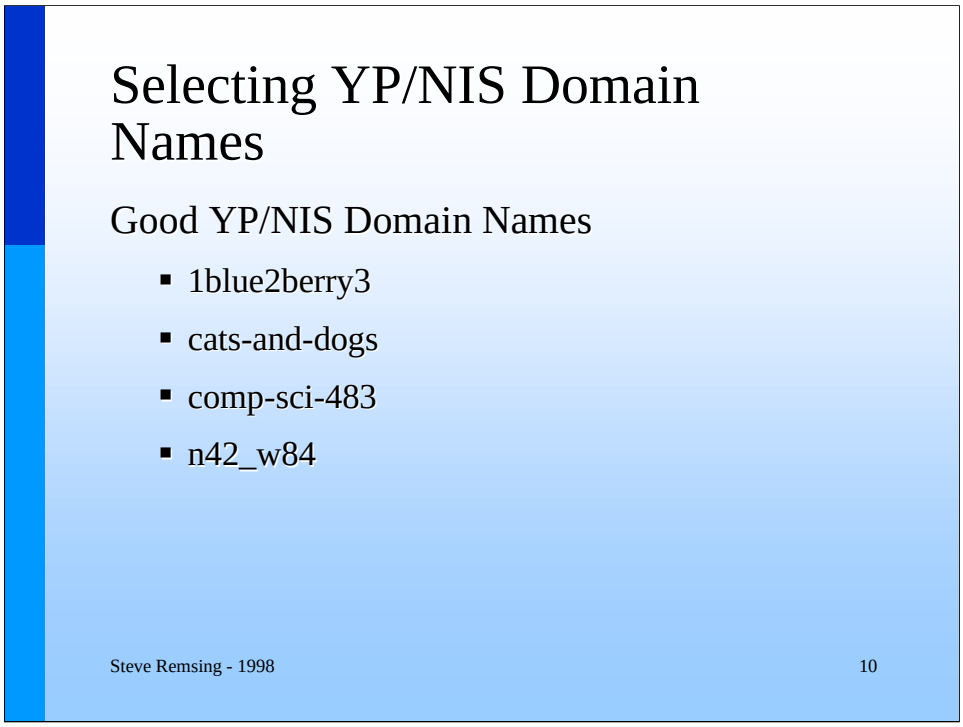
ftp://coast.cs.purdue.edu/pub/tools/unix/ypx.shar



Notes:

Why are these bad?

- oak - This is the name of the server and the first name YPX will use as a guess.
- oak.acorn.com - This is the FQDN of the server, the second name YPX will use as a guess.
- acorn.com - This is the Internet domain name, also easy to guess.
- acorn - This is the name of the organization, a likely candidate that is easy to guess.
- acorn-furniture - Again, this is the name of the organization.



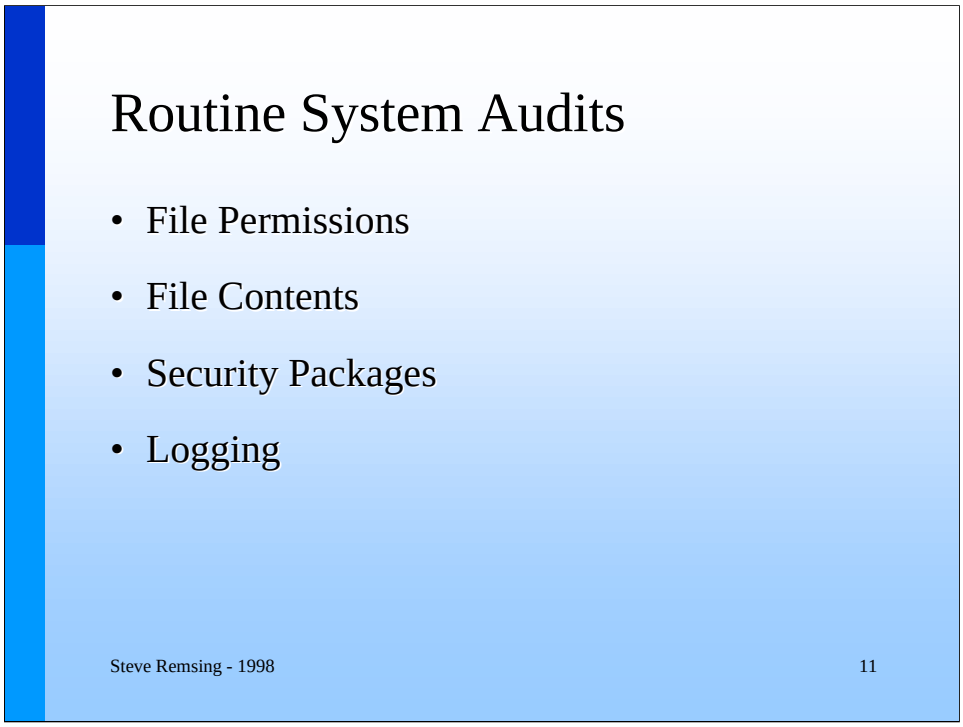
Notes:

Any of the above would be a good domain name for Acorn because they are all unrelated to the hostname, organization name, oak trees, acorns, and furniture.

Since the domain name is only used by the system administrator during the initial installation some people suggest using a domain name like the following:

a_long_string_of_words_that_have_no_meaning

I consider this overkill, but do what you think best.



Notes:

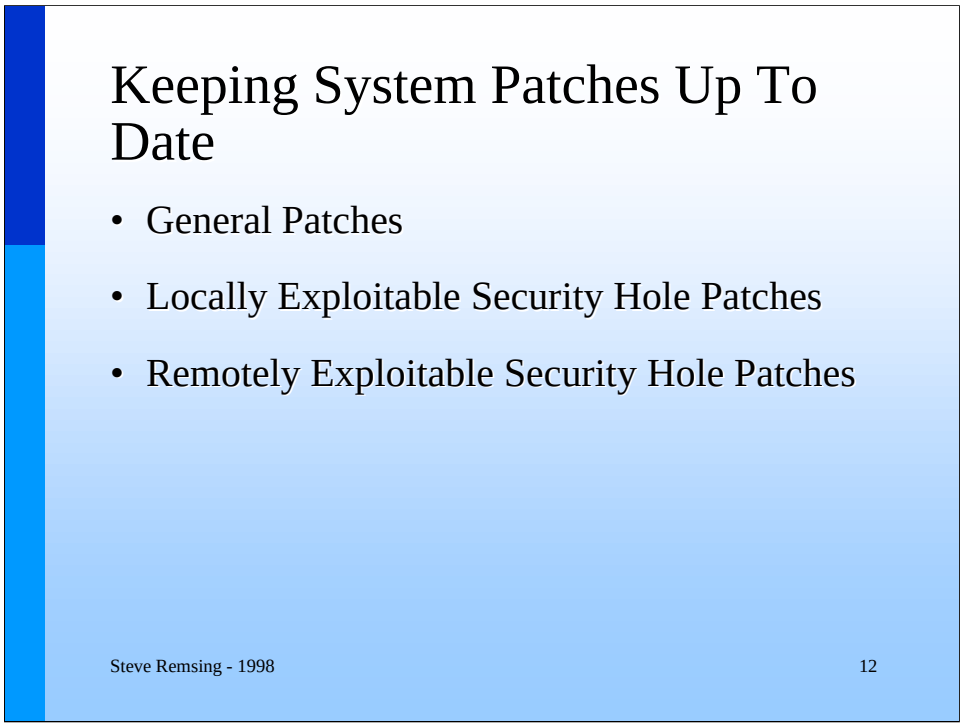
Find your mistakes before the hackers do, you will save on aspirin.

Never assume that since the machine was installed correctly that its configuration is still correct. What if it wasn't installed correctly? It is possible that a step was missed in the installation process that left a gaping hole in the system.

Also, often times system administrators will make a change to the system "just to test something" and forget to put the files back to the correct state. This can create security holes.

Network or server configuration changes often require subsequent changes on all machines on the network. It is possible, especially in a large environment, that a machine or two will be missed when making these changes.

Not only will audits help you improve your security for the future, they may also uncover an existing break-in.

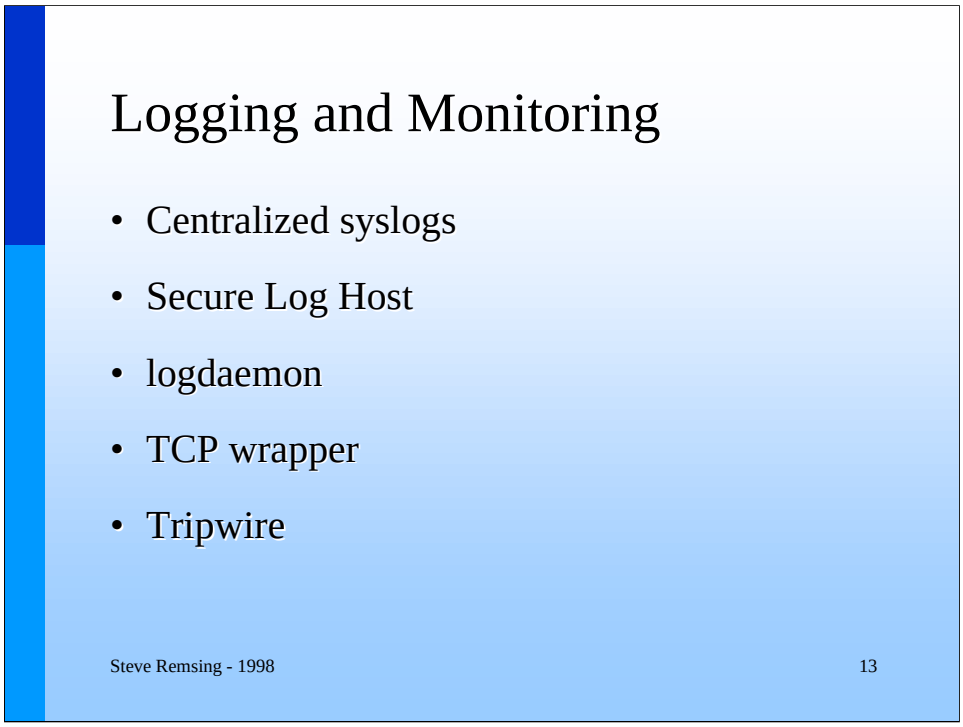


Notes:

In general, installing vendor patches is a good idea. Installing non-security related patches is optional but you should consider all security related patches mandatory.

There are two classes of security holes: locally exploitable and remotely exploitable. Locally exploitable holes require login access to the machine and therefore some people consider them less of a threat. Remotely exploitable holes do not require login access and can be exploited from any system. They are clearly a very serious threat.

Any remotely exploitable hole must be patched as soon as possible. Many people tend to overlook the need to patch locally exploitable holes in a timely manner. Do not fall into this trap. Consider all security patches a high priority regardless of the nature of the exploit.

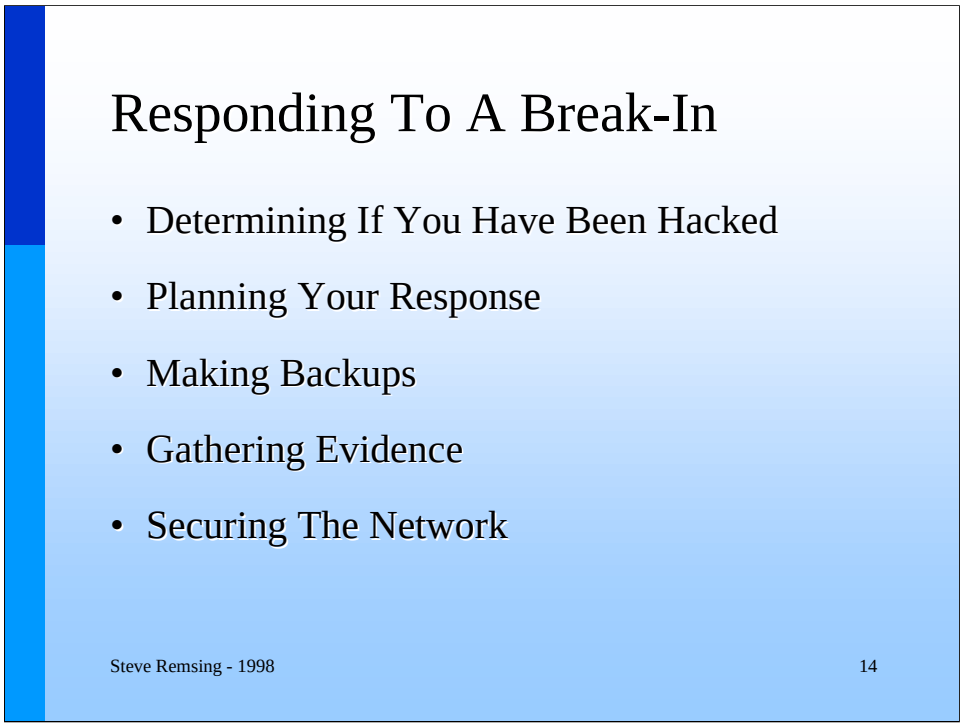


Notes:

Centralized logging is essential to being able to effectively investigate a security incident. Without it each machine must be checked locally, a time consuming process. Having one central log file also makes it much easier to see the entire incident on a timeline. Centralized logging requires a *very large* disk.

Centralized logging also allows you to design a secured log host to receive a copy of all syslog messages. Ideally, this secured log host will be a dedicated machine with no other function and very limited access. Remember, you can not trust the logs on a compromised system. A secured central log host is the only way to have any confidence in the validity of your log files.

I suggest you configure your system to store all logs on a separate disk partition that is only used for logging. This will help prevent a DoS attack if someone tries to flood your logs.



Notes:

When responding to an incident, first and foremost you must follow your organization's policy regarding such issues, in particular when working on the system and discussing the situation with others.

Notes:

Often times the most difficult part of investigating a possible security incident is determining if you have really been hacked, or if there is an innocent explanation for whatever first made you suspect a break-in.

By carefully investigating the system configuration and all logs you should be able to determine if you are truly dealing with a break-in or not. If after looking into the situation you are unsure, play it safe and treat it like a break-in.

Notes:

One of the first places to look for signs of a break-in is the system's "last log". In the output of the `last` command you should look for any unusual logins.

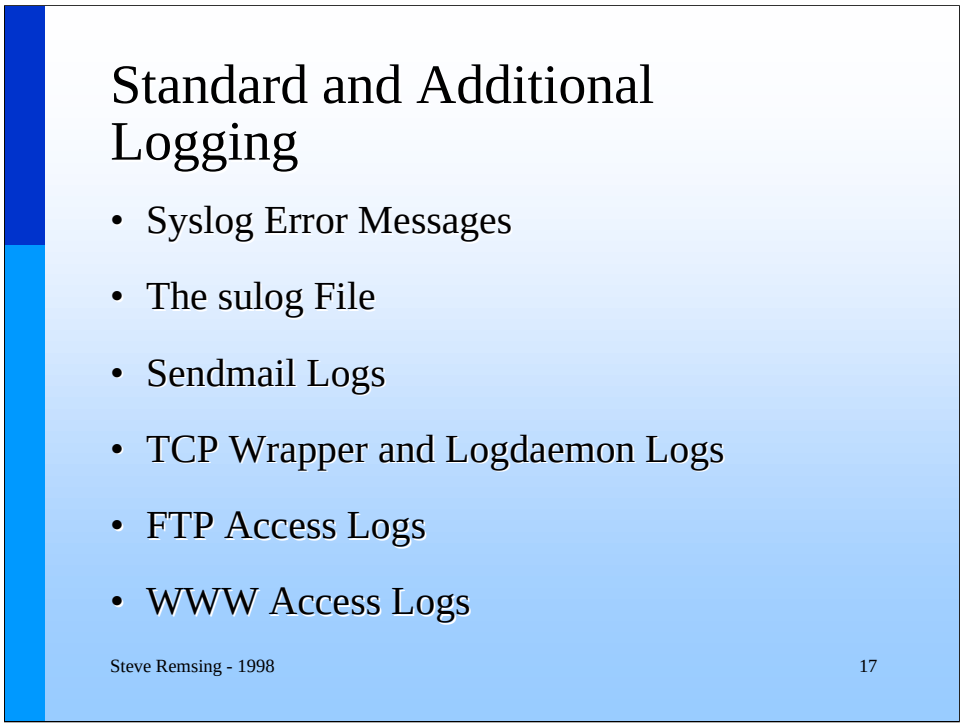
The last log will show who logged in, what terminal they logged in on, what time they logged in, how long they were on the system, and where the login came from. If you have never used the `last` command, read the man page on it today.

As helpful as the last log is, there is one fatal flaw: it can be tampered. If the person was able to gain root access, they can alter the file where the login data is stored. All is not lost however. While it won't help you identify where the attack came from, when it occurred, or the initial account used to gain access, there are signs that will let you know the files have been tampered.

Truncated files that only contain recent data, files that have large periods of time without data, duplicated login records, or zeroed out records are all signs that someone has tampered with the files.

There are many free tools to help you determine if your last log files have been altered, one of them is chkwtmp. It is available at:

ftp://coast.cs.purdue.edu/pub/tools/unix/chkwtmp/



Notes:

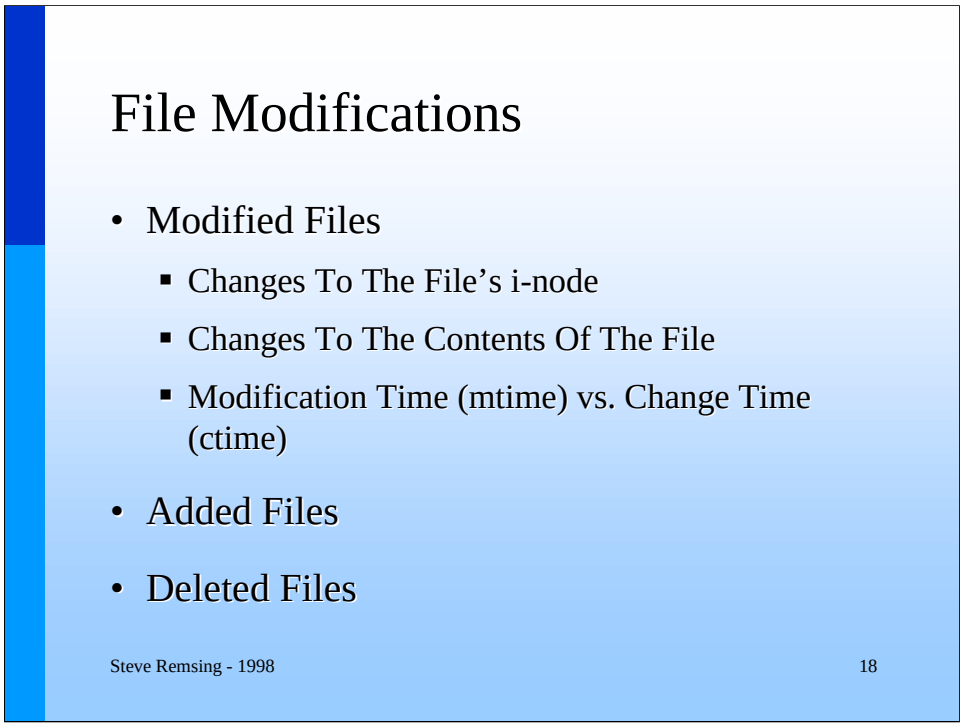
After looking at the last log data you should review the other logs you have available, looking for any similarities or differences. Every system has the standard syslog messages (often in /var/adm/messages), a record of the `su` command (/var/adm/sulog), and sendmail logs (sometimes included with the standard syslog messages).

Additionally, if the system is running as a WWW or FTP server you should have access logs for these services.

Lastly, you may have additional logging from security packages such as the TCP wrapper or logdaemon packages.

All of these logs are valuable in determining who accessed your system, when they did it, and where the connection came from. The TCP wrapper software can even be configured to finger the remote system for a record of who was logged on at the time.

All these logs have the same basic problem as the last log: if someone gains root access to the system with the logs, they can modify them to remove all traces of their connection. To be useful and reliable, all logs should be sent to a secured log host that has no other function and very limited access.



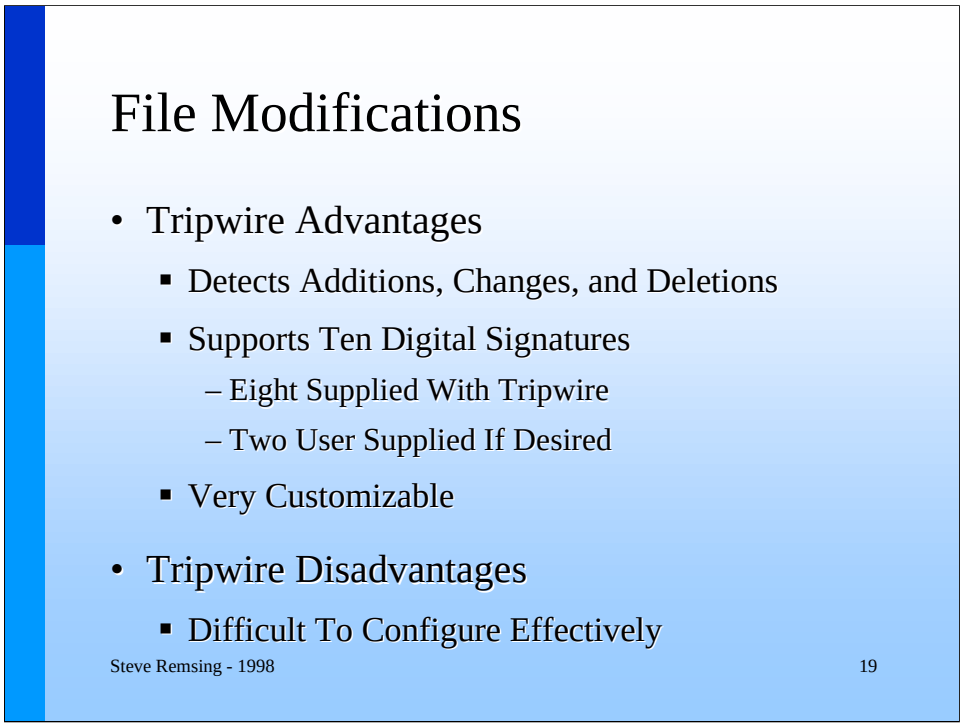
Notes:

A sure sign that your system has been compromised is unexplained file modifications, especially to system configuration files or system binaries.

There are two main things to check to determine if a file has been modified. The first is to look for any changes to the i-node information for the file; such as the owner, the group, the permissions, number of links, etc. Second, you look for changes to the contents of the file. This is only useful for small text files, such as /.rhosts or /etc/passwd, where you can easily notice a change.

Detecting these changes can be difficult and you don't want to look at every file on your system. To only look at files that have been modified recently you can check the modification time (mtime) of the file (shown by `ls -l`) and only look at files with recent timestamps. However, the mtime is only updated if the file contents change, it will not show changes to the i-node, and the mtime can be falsified using the `touch` (see the man page) command. The ctime (shown by `ls -lc`) is harder to falsify and it is updated with changes to either the i-node or the file contents.

A file being added or deleted should also be a concern. New files could either be designed to open a security hole or they may be sniffer log files, especially if they are located in odd places. Deleted files may be an attempt to cover up the record of a break-in or remove security control files.



Notes:

Checking for file modifications by hand is an impossible task. Fortunately there is a tool to help: Tripwire.

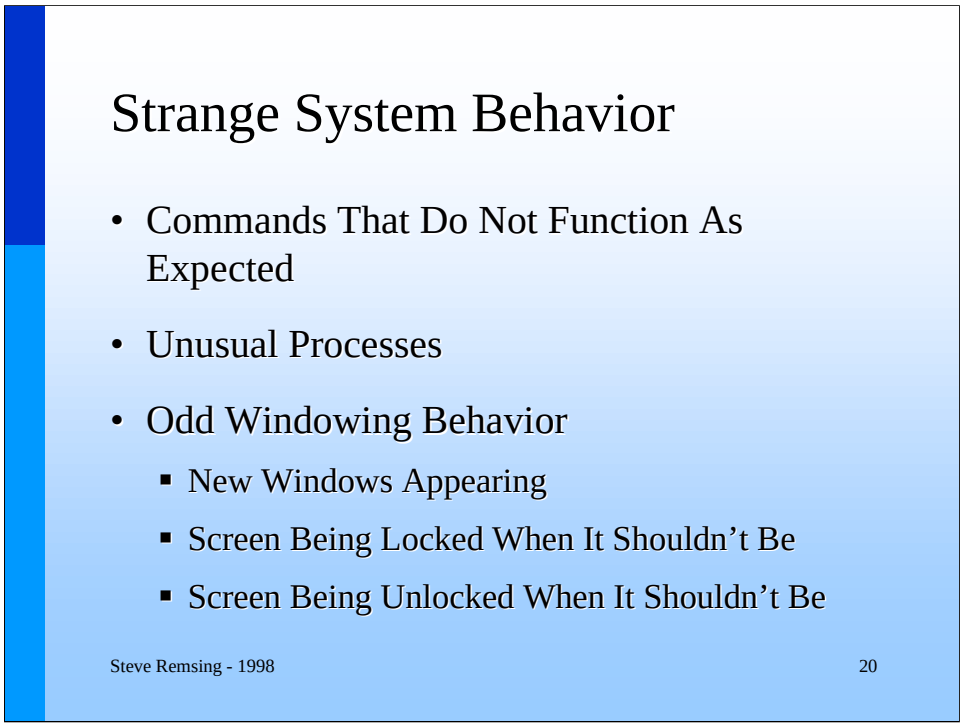
Tripwire takes a snapshot of your current system, including all the i-node information and up to ten digital signatures of every file on the system during the initialization phase. Once you have this initial snapshot, each subsequent Tripwire run will compare the current values with those stored in the database and report any differences, including added or deleted files.

Tripwire is very customizable, you can select what files are included in the Tripwire database and what level of modification is acceptable before it should be reported. For example, you shouldn't have Tripwire report changes to the size or contents of your log files but you should report ownership changes or if the file is deleted.

Getting the configuration file that tells Tripwire what files to look at and what to changes report correct is a difficult task. The configuration is very specific to each different version of UNIX and to your site.

Some sample configuration files come with Tripwire. I also have a generic version of the files I use posted at:

http://lheawww.gsfc.nasa.gov/~srr/tripwire.html



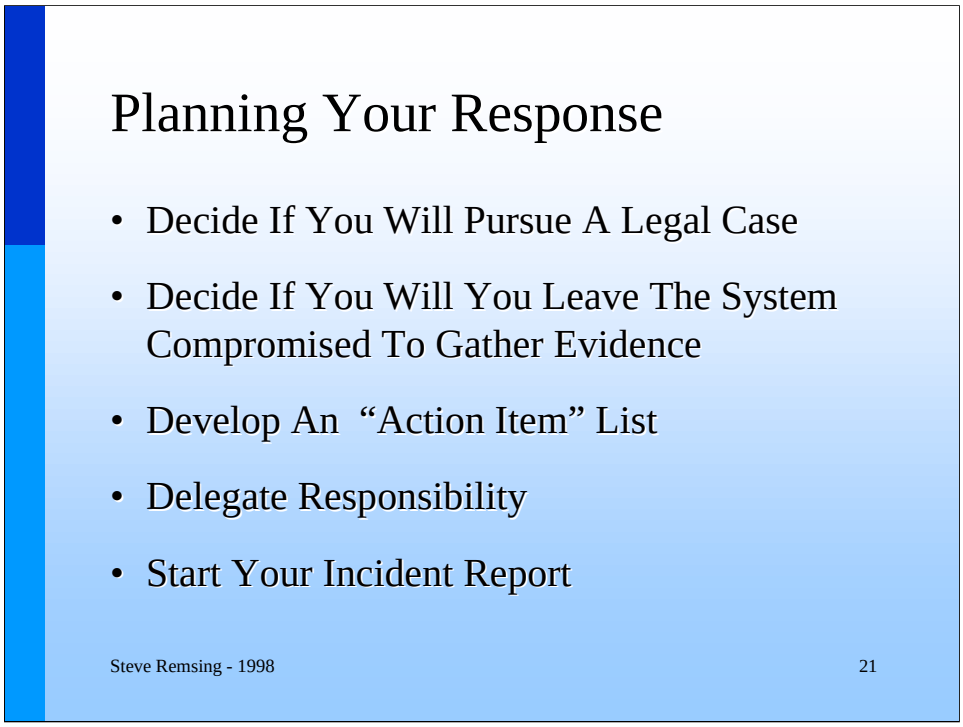
Notes:

Another sign that your system has been compromised is strange system behavior. Strange behavior is hard to define but you will likely know it when you see it.

Hackers often will replace system binaries with trojan versions, usually this trojan version will perform exactly like the original command from the user's point of view but not always. If a system command suddenly fails to work as expected you should be concerned.

Also, many times a hacker will leave processes running on your system such as a sniffer or password cracker. If you notice strange processes that have accumulated a large amount of CPU time or if you notice unusually poor system performance you should investigate.

Lastly, there are a number of tricks intruders will play with X windows such as recording keystrokes, opening a new window, locking your screen with a false screen lock, etc. Most of these attacks are designed to capture passwords. Be very careful with your use of the `xhost` command.

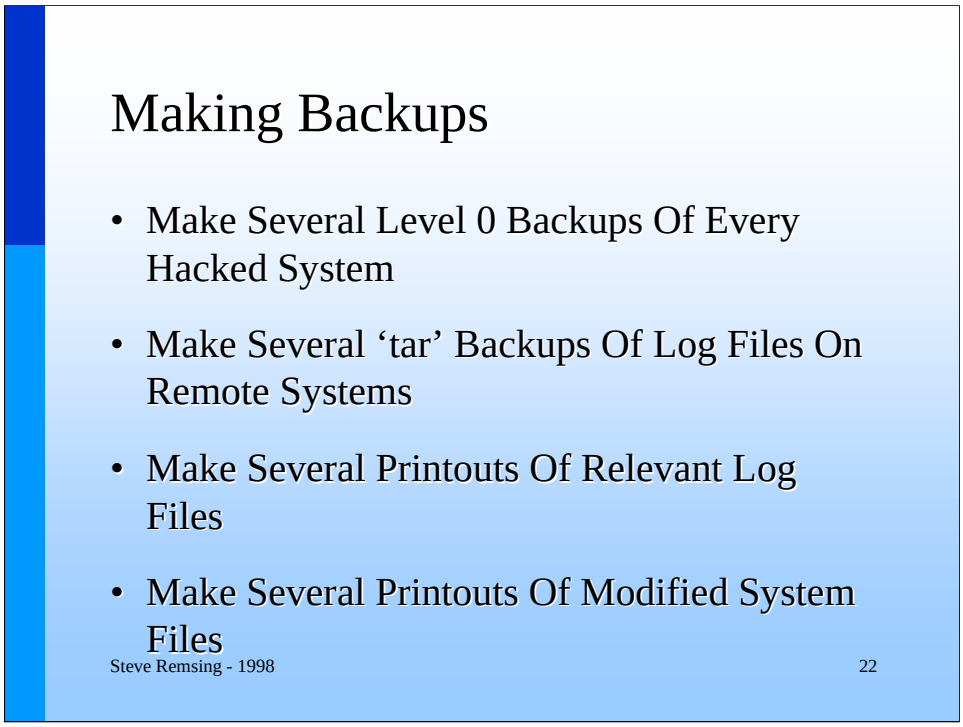


Notes:

Planning is the key to a successful investigation. Without careful planning mistakes will be made. You may overlook critical evidence, you may destroy evidence, or you may not close all the holes on your system.

Before you can begin, you must decide if you plan to pursue a legal and decide if you will leave your system compromised to gather more evidence. If your answer is yes to either you must be more careful with your investigation.

Once you know your position on these two issues you need to develop an action item list of what must be done to achieve your goals. To ensure that all the work is accomplished you should delegate the workload to all admin personal. This avoids the problems of duplication of work and ensures all the work is performed. At this point you should also begin your incident report by documenting what is to be done and who is responsible for doing it.

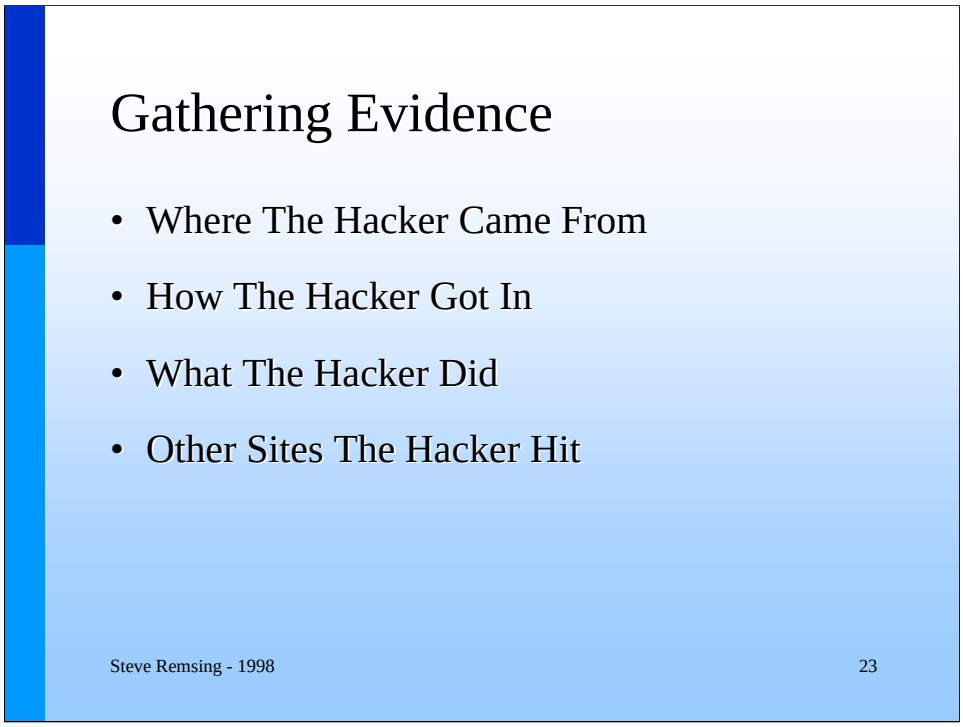


Notes:

Backups of any compromised system are a must. Always make several level 0 dumps of every filesystem on a compromised machine. If you pursue a legal case the law enforcement officials will require at least one copy. You should also have a copy for yourself.

Make additional backups of other relevant log files that are stored on remote systems as well. It is also helpful to print the relevant files or sections of log files that show either the log of connections during the break-in or the changes made to the system. Again, make several tape backups and printed copies because you will need to provide at least one to law enforcement.

The copy of the tapes and printouts that you keep are useful in determining what exactly was done to the system and possibly how they gained access. You can restore the full backup of the system to a large unused partition and explore it in detail while the system itself has been re-installed and put back into use. This allows you to investigate the incident fully without increased downtime.

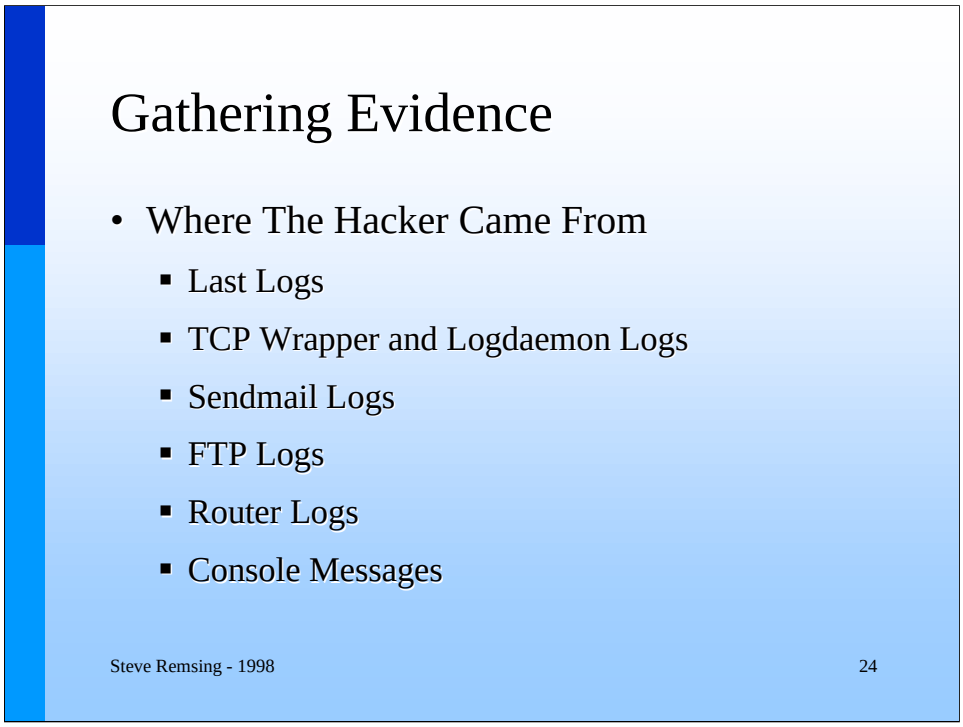


Notes:

When gathering evidence you should try to answer the questions above. This will show that the machine was compromised and lead law enforcement toward the person responsible. Even if you don't want to pursue a legal case, answering these questions is helpful in preventing future break-ins because they show you where you need to improve your security.

While determining a list of other sites that the hacker attacked (whether from your systems or not) will likely not help you directly improve your security, it is important to contact these sites and let them know they may have been compromised.

By sharing such information we can improve the overall security of machines on the Internet, which will benefit everyone.



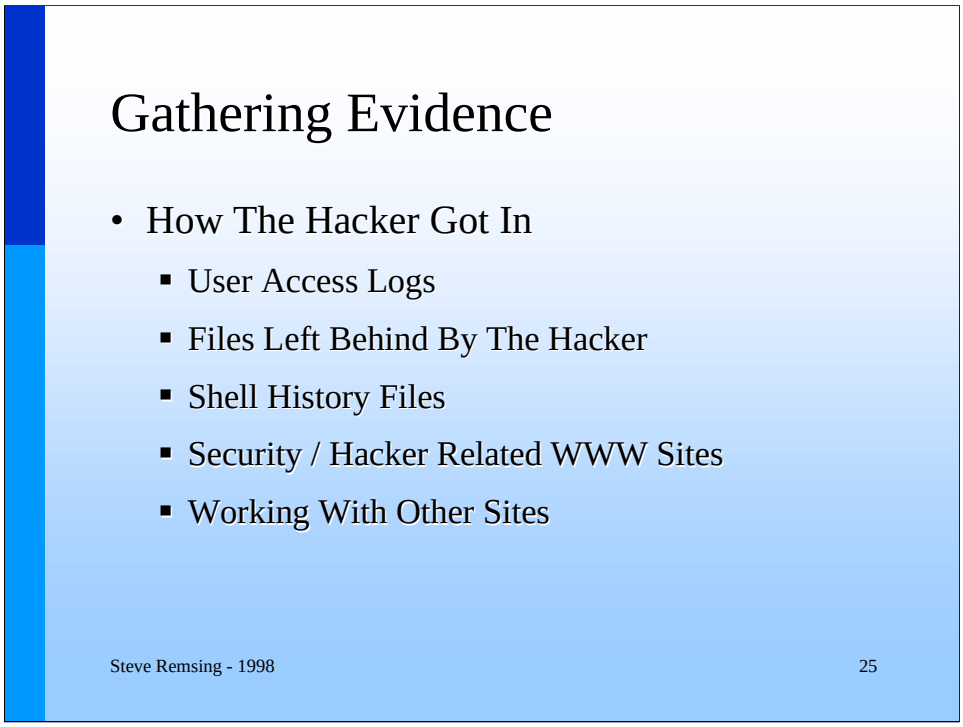
Notes:

Your site may have additional logs beyond those listed above, such as firewall logs, modem use logs, or process accounting logs. Look at and compare all sources of log information.

Again, logs are only useful if you can trust that they have not been altered. At least some of your logs should be a secured log host.

A nice package to help you manage your log files is swatch. This package is available from a number of sites, including:

ftp://coast.cs.purdue.edu/pub/tools/unix/swatch



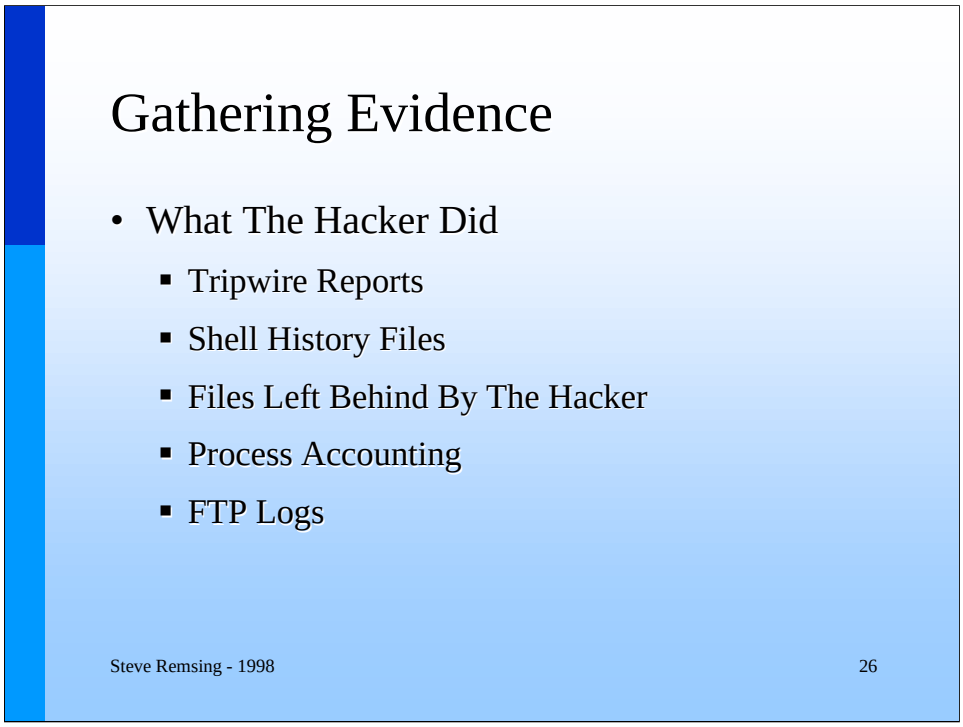
Notes:

One of the most important aspects of the investigation is determining how the hacker gained access to your system in the first place. This is the first hole you need to close. Most root compromises start with gaining user level access since that is much easier. You must determine how they gained user access and then how they gained root access.

If you know of files left behind by the hacker or shell history files showing commands the hacker issued, these may help you determine how they gained root access.

Security/Hacker related web sites are also great sources of information, especially if you have an idea of the method used to gain access.

Lastly, you can try working with other sites that have been hit by the hacker, as seen in your logs, or the site where the attack came from. Be careful when contacting the site where the attack came from, for all you know the admin for that site is the hacker…



Notes:

There are a number of sources of information that might reveal what the hacker did while on your system. Unfortunately, a careful hacker will most often remove or destroy many of these sources.

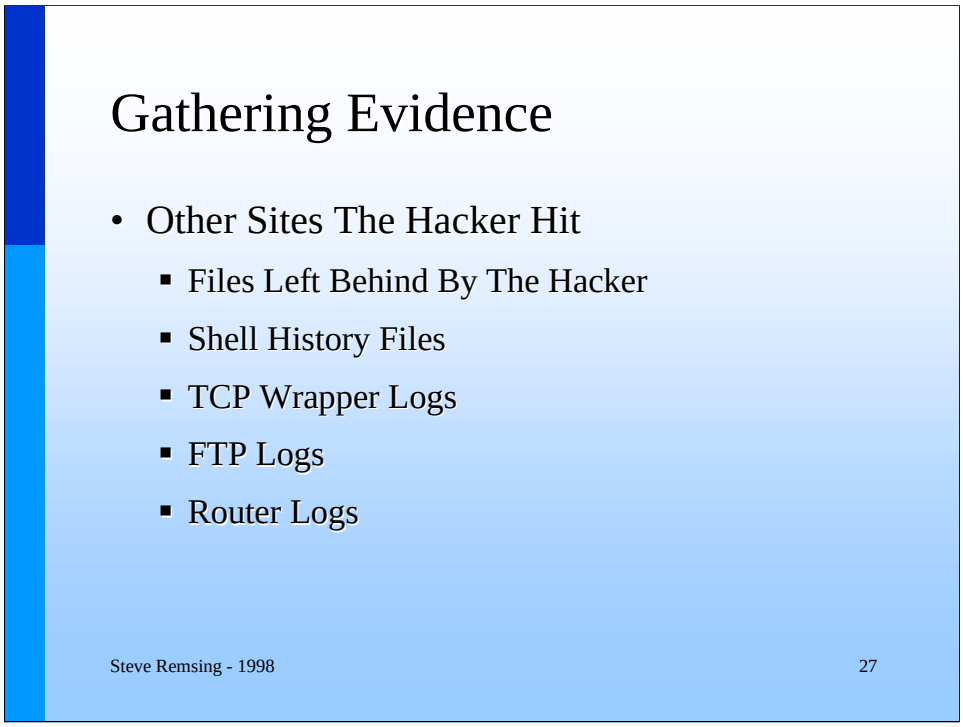
Tripwire reports, if logged on a secure host, are an excellent source of information. These reports can't always tell you what the changes were but they at least let you know where to look.

If you are lucky, the hacker will leave behind a shell history file showing all the commands issued on your system. Intruders also may leave files in what they hope is a hidden directory. Often these files are programs or scripts used to gain additional access, clean the system logs, or leave back doors on the system to future use.

Process accounting logs, if you have them, may also give you some insight into what was done on your system.

Detailed FTP logs may show the names of any files the hacker put on the system or copied off the system. If files were added to your system and later removed try a web search on the filename, you are likely to turn up something.

Knowing the tools used by the hacker helps you understand what was done on your system and to defend against future attacks.



Notes:

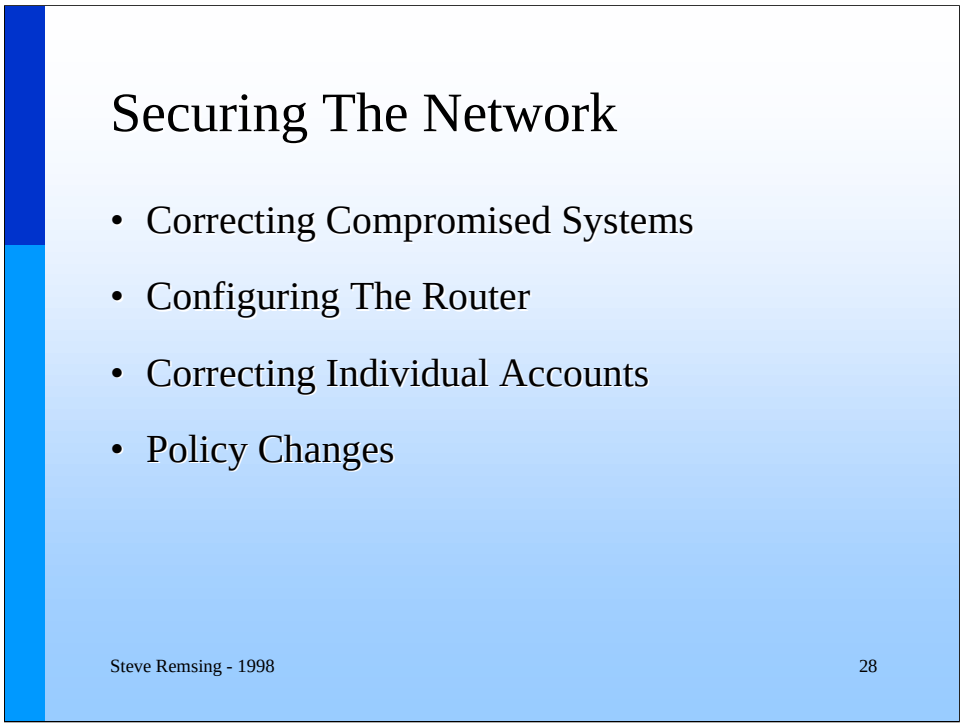
If you can identify other sites that the hacker attacked, it is important to inform the administrators of those sites.

Evidence that other sites were attacked will be found in the same files you have been looking at already. The two most likely sources are files left behind by the hacker (maybe they left a target list?) and shell history files. There are other sources that you may not have thought about.

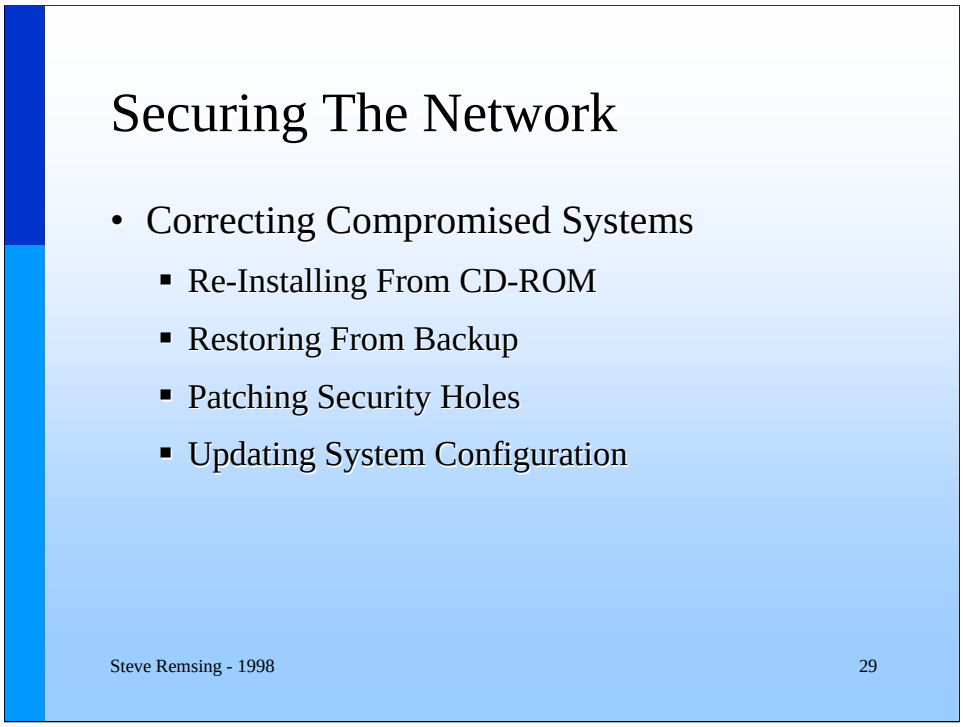
TCP wrapper logs may reveal other sites the hacker has attacked if the remote sites is using the TCP wrapper also. Most sites that use the TCP wrapper configure it to finger the machine where the connection originates. If you log finger requests on your systems you will see finger requests from the site under attack each time the hacker tries to connect to their systems.

FTP logs may show files being transferred to other sites, which may indicate that the remote site may have been attacked.

Router logs, if you have them, should show all connections, both incoming and outgoing. This information will tell you exactly what sites were visited and the types of connections made to them.


Securing The Network
• Correcting Compromised Systems
• Configuring The Router
• Correcting Individual Accounts
• Policy Changes
Steve Remsing - 1998
28

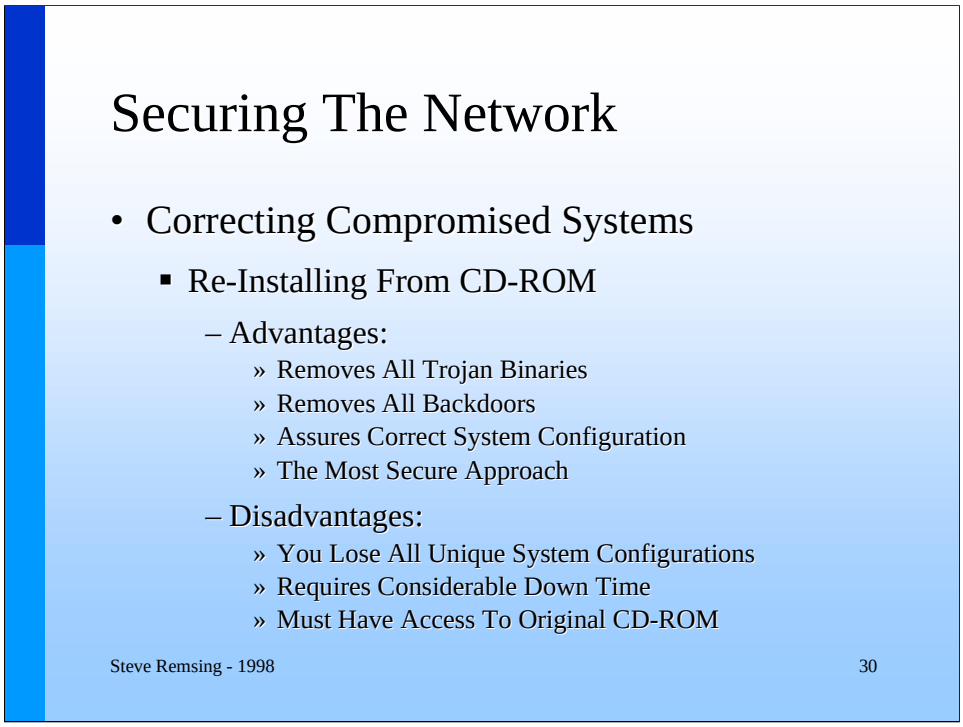
Notes:



Notes:

It is not enough to re-install or restore from backup the contents of a compromised system. Doing so will leave the same holes that the hacker used to break into the system.

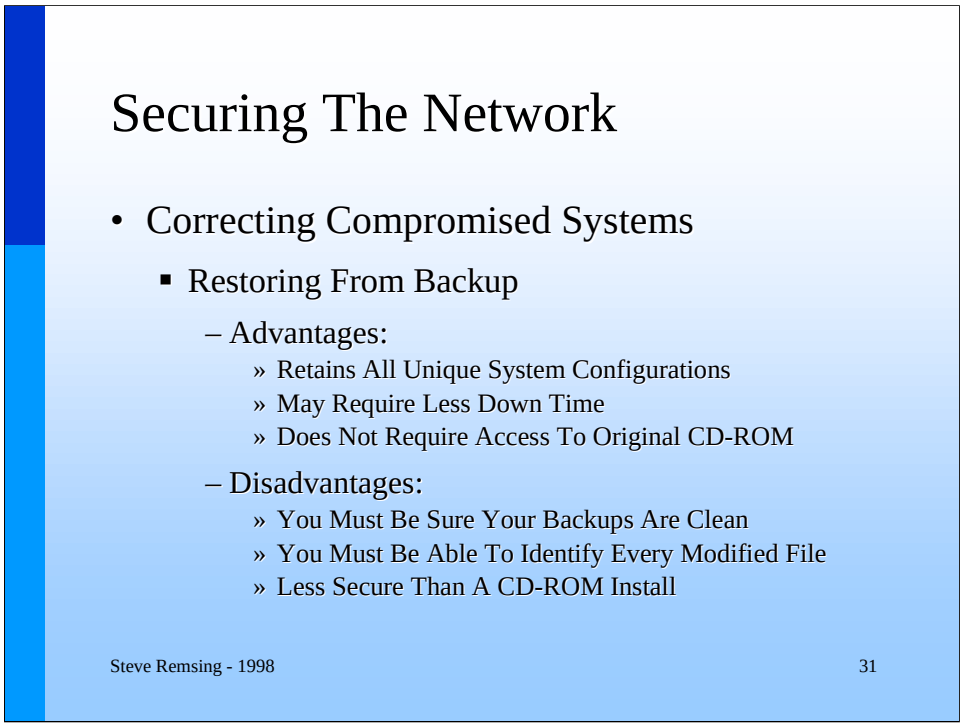
After either re-installing or restoring your system you must patch any security holes that were identified during the investigation or correct any system configuration issues that created any vulnerabilities on the system, for example: NFS exports, trusted hosts, or inetd services.



Notes:

If you have a record, such as a Tripwire report, of all the files that were modified, added, or deleted you may prefer to restore your system from your backups instead of re-installing from CD-ROM.

Unless you don't have access to the original CR-ROM or have a very unique customized configuration that would be very difficult to reproduce, I would ***strongly suggest*** re-installing from the CD-ROM.

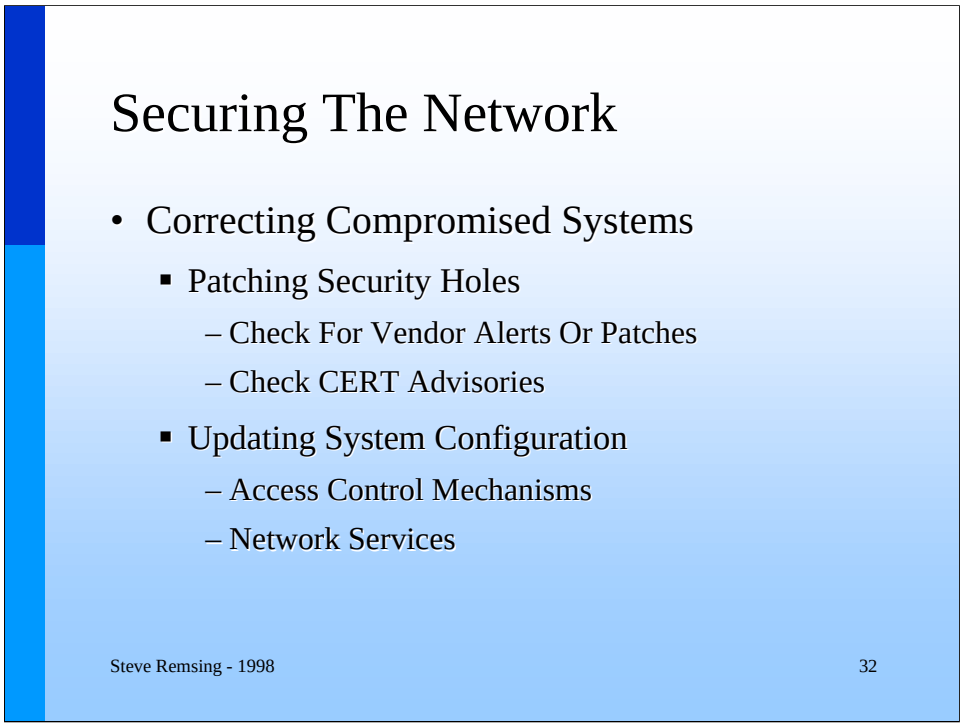


Notes:

If you are completely confident that your backups are clean and you are certain you know everything that was changed on your system you may chose this method.

I suggest you only consider it if you had a very small break-in by an obviously inexperienced hacker. Restoring from tape may be faster and easier but it is not worth the risk of leaving a hole on your system.

Some sites go so far as to reformat all the disks attached to a compromised machine, only restoring data partitions. Depending on your organization and the nature of the break-in, maybe this is justified.



Notes:

After re-installing or restoring your system, search your vendor's patch list and review all recent CERT or other such advisories. It is unlikely that your machine was compromised by a new method.

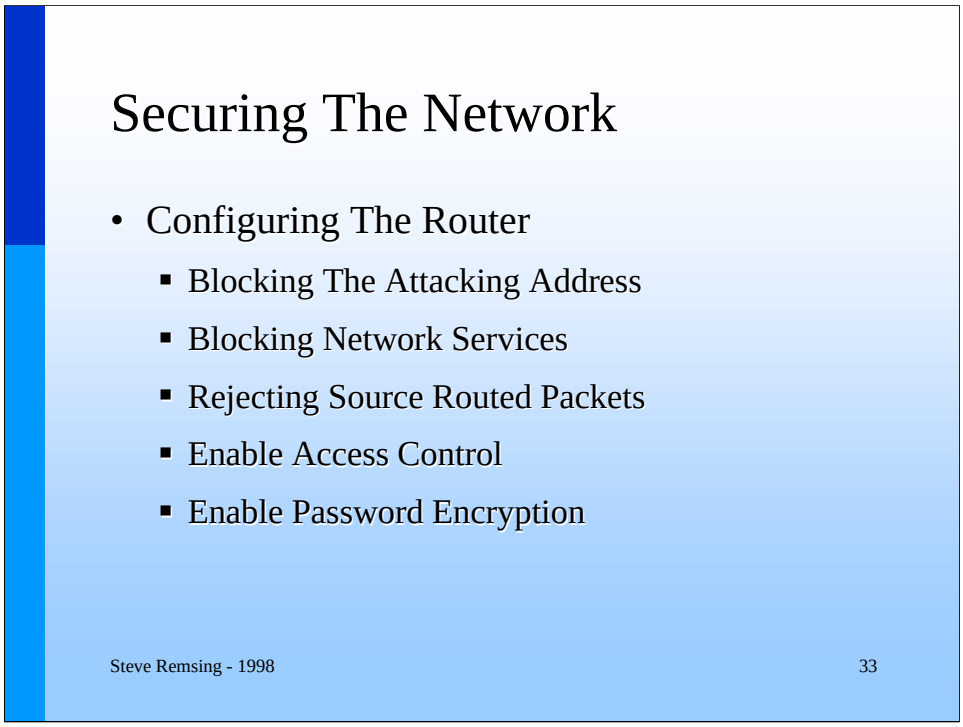
This would also be the time to correct any system configuration issues that might be exposing your machines, such as:

/.rhosts file
/etc/hosts.equiv file
NFS exported filesystems
Open file permissions
Poor WWW or FTP configuration

Use tools such as COPS, SATAN, ISS, and Crack to find your weak points.

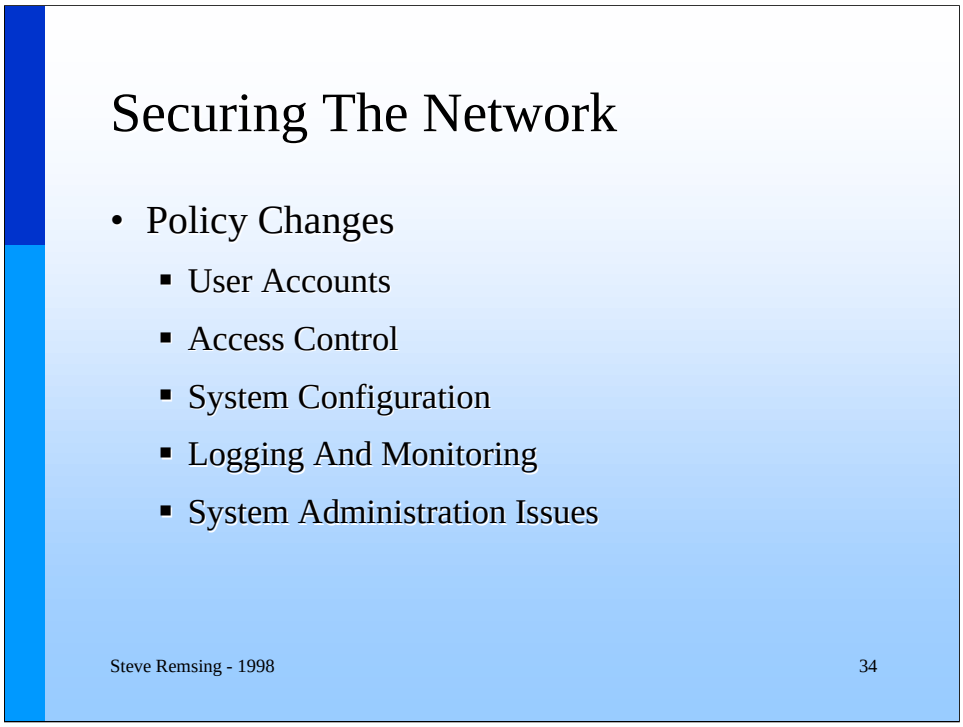
All these tools can be found at:

ftp://coast.cs.purdue.edu/pub/tools/unix/



Notes:

When looking at your site's security, don't forget to look at your router configuration, especially if you have a firewall configuration. If someone can successfully attack your router it will be much harder to prevent a successful attack on your machines, not to mention the possible denial of service aspect of someone else having control of your router.

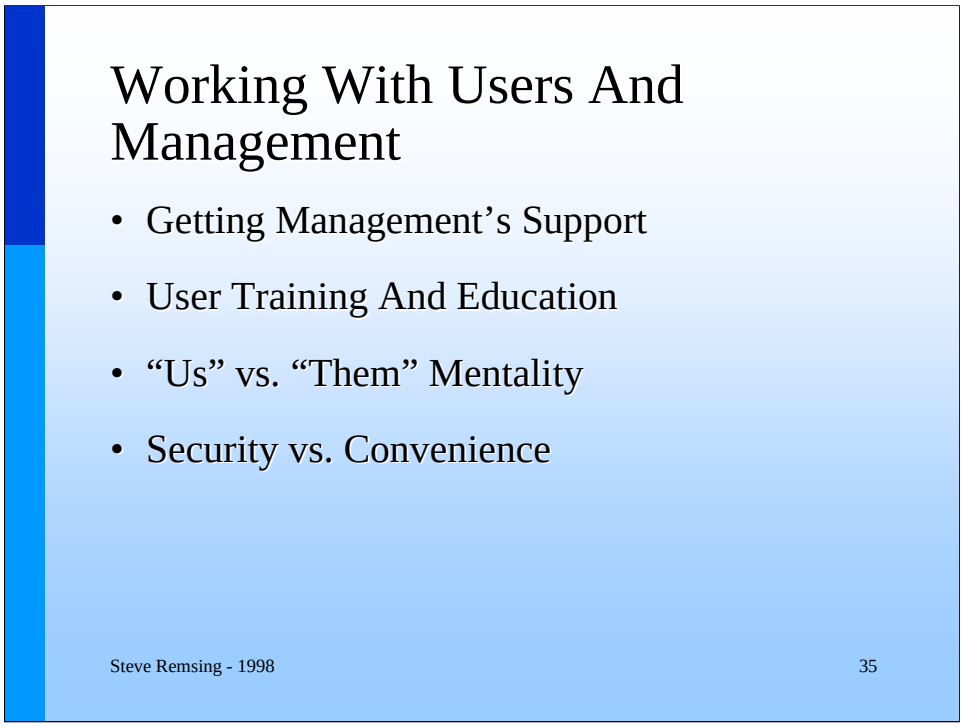


Notes:

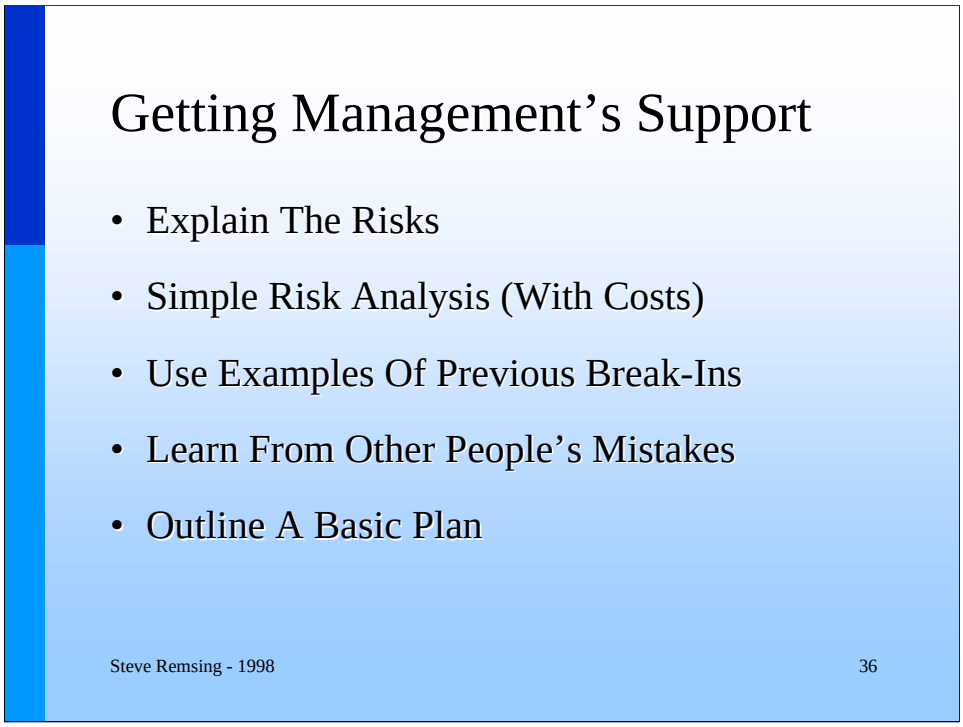
Use the break-in to your advantage. After you have the situation under control you can propose policy changes that would improve your security.

If you can demonstrate how the policy change would prevent a future attack you have much better chance of getting people to go along with it.

For example, if your current policy allows remote access using regular reusable UNIX passwords and someone broke in by sniffing a user's password, you might be able to change the policy to require all remote access logins to use one time passwords.


Working With Users And Management
• Getting Management's Support
• User Training And Education
• "Us" vs. "Them" Mentality
• Security vs. Convenience
Steve Remsing - 1998
35

Notes:



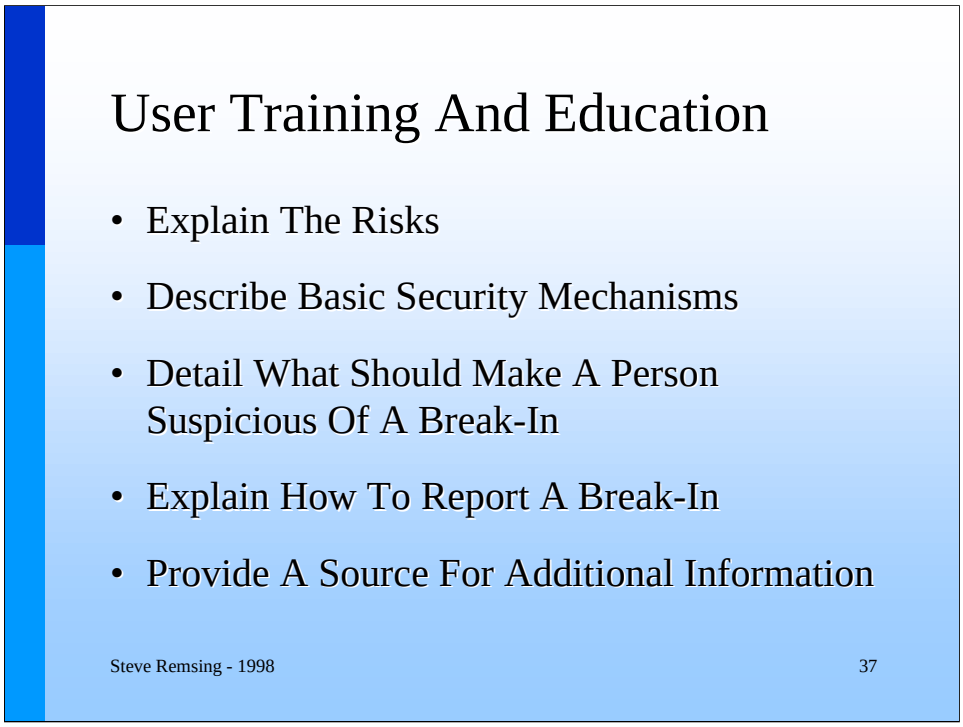
Notes:

In general, management does not understand the technical issues surrounding security. They understand risk management and analysis.

Explain the cost of a security incident in terms of the number of hours spent investigating the situation, the downtime while the system is recovered, loss of data, etc.

There have been a number of break-ins that have made headlines, use these examples: Department of Justice, CIA, Air Force, NASA, and Yahoo to name a few.

Present a reasonable solution, with an estimation of cost and user impact that addresses the major concerns you face.

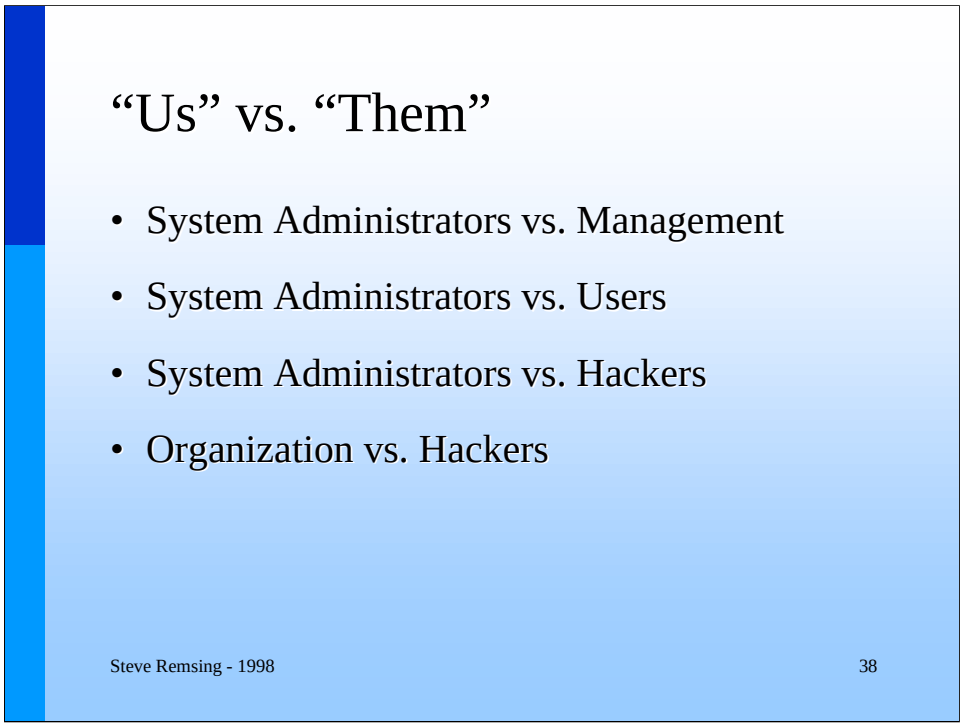


Notes:

As with management, explain the risks of a break-in to the users in terms they can relate to: data loss, data corruption, and downtime. If a hacker corrupts the source code of a major product with a release date of two weeks, the programmers are going to be putting in some overtime.

Most users are interested in security issues, they just don't have the information they need. User training in computer security can have a very positive impact if you present it as ways users can improve security and help the admin staff instead of a list of "don't do this" items. Computer security shouldn't come across as medicine.

Have a suggested reading list of papers, web pages, and books ready for anyone who is interested.

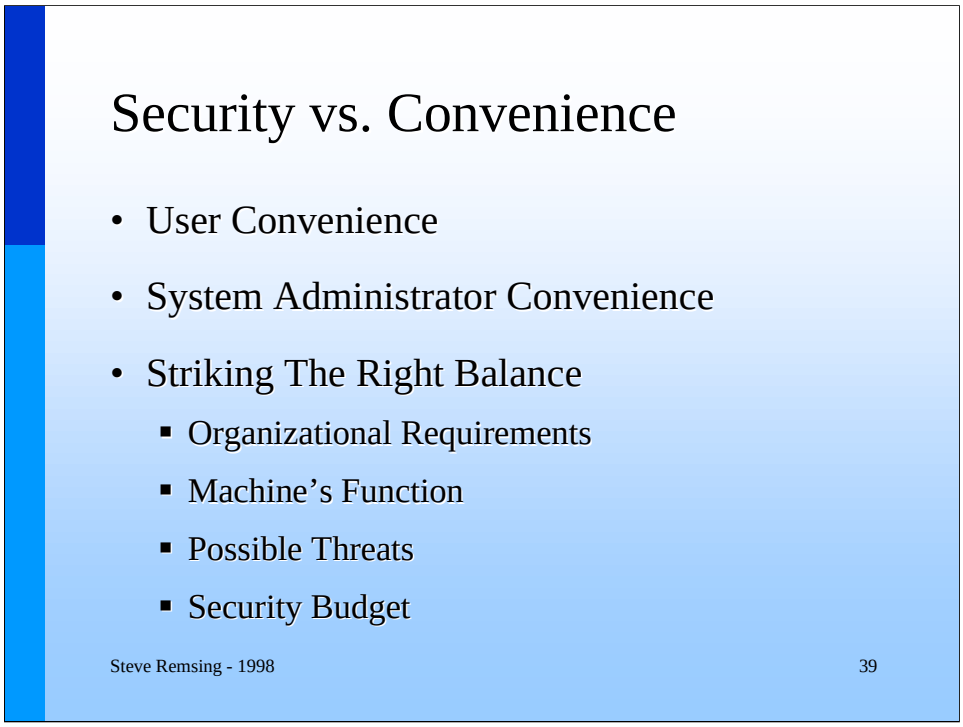


Notes:

Working with management and users, you can turn the "Us" vs. "Them" mentality in your favor.

If the users don't support your security actions you will have an uphill battle. If management doesn't, you have already lost the war.

If management holds you responsible for any security incidents but does not support your actions to improve security, update your resume and find a new job now before you are forced to.



Notes:

Every site must decide the appropriate level of security for their needs. There is no one-size-fits-all solution.

Even within a given organization, different classes of machines may have different security needs.

If your machines are configured to be reasonably difficult to exploit without being overly inconvenient for either the users or the admins you likely have a good balance for your site.

No matter where your balance point is, **log everything.**

Remember:

Just Because Your Paranoid Doesn't Mean They Aren't Out To Get You.

Notes:

If you have any questions, comments, or would just like to discuss any of these topics in more detail send mail e-mail at:

srr@lheamail.gsfc.nasa.gov